Abstract

Artificial Neural Networks (ANNs) are algorithmic techniques that simulate biological neural systems. Typical realization of ANNs are software solutions using High Level Languages (HLLs) such as C, C++, etc. Such solutions have performance limitations which can be attributed to one of the following reasons:

- Code generated by the compiler cannot perform application specific optimizations.
- Communication latencies between processors through a memory hierarchy could be significant due to non-deterministic nature of the communications. In data mining _eld, ANN algorithms have been widely used as classifiers for data classification applications. Classification involves predicting a certain outcome based on a given input. In order to predict the outcome more precisely, the training algorithms should discover relationships between the attributes to make the prediction possible. So later, when an unseen pattern containing same set of attributes except for the prediction attribute (which is not known yet) is given to the algorithm it can process that pattern and produce its outcome. The prediction accuracy which defines how good the algorithm is in recognizing unseen patterns, depends on how well the algorithm is trained. Radial Basis Function Neural Network (RBFNN) is a type of neural network which has been widely used in classification applications. A pure software implementation of this network will not be able to cope with the performance expected of high-performance ANN applications. Accelerators can be used to speed-up these kinds of applications. Accelerators can take many forms. They range from especially configured cores to reconfigurable circuits. Multi-core and GPU based accelerators can speed-up these applications up to several orders of magnitude when compared to general purpose processors (GPPs). The efficiency of accelerators for RBFNN reduce as the network size increases. Custom hardware implementation is often required to exploit the parallelism and minimize computing time for real time application requirements. Neural networks have been implemented on different hardware platforms such as Application-Specific Integrated Circuits (ASICs) and Field Programmable Logic Gate Arrays (FPGAs). We provide a generic hardware solution for classification using RBFNN and Feed-forward Neural Network with backpropagation learning algorithm (FFBPNN) on a reconfigurable data path that overcomes the major drawback of _axed-function hardware data paths which offers limited edibility in terms of application interchangeability and scalability.

Our contributions in this thesis are as follows:

- Deification and implementation of open-source reference software implementation of a few categories of ANNs for classification purpose.
- Benchmarking the performance on general processors.
- Porting the source code for execution on GPU using Cuda API and benchmarking the performance.
- Proposing scalable and area efficient hardware architectures for training the learning parameters of ANN.
- Synthesizing the ANN on reconfigurable architectures.
- MPSoC implementation of ANNs for functional verification of our implementation
- Demonstration of the performance advantage of ANN realization on reconfigurable architectures over CPU and GPU for classification applications.
- Proposing a generalized methodology for realization of classification using ANNs on reconfigurable
architectures.